(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
3 April 2003 (03.04.2003)

PCT

(10) International Publication Number
WO 03/027891 A1

(51) International Patent Classification[7]: G06F 15/173

(21) International Application Number: PCT/US02/31205

(22) International Filing Date:
30 September 2002 (30.09.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/326,023 28 September 2001 (28.09.2001) US

(71) Applicant: **COMMVAULT SYSTEMS, INC.** [US/US]; 2 Crescent Place, Oceanport, NJ 07757-0090 (US).

(72) Inventors: **PRAHLAD, Anand**; 3 Bucknell Drive, East Brunswick, NJ 08816 (US). **MAY, Andreas**; 1 Carter Drive, Marlboro, NJ 07746 (US). **WANG, Zhao**; 10 Kroeger Lane, Piscataway, NJ 08854 (US). **DEMENO, Randy**; 58 Elmbank Street, Staten Island, NY 10312 (US). **IYER, Tinku**; 4100 Ponytail Drive, Apt. 506, Mississauga, Ontario L4W 2Y1 (CA).

(74) Agent: **OSTROW, Seth, H.**; Brown Raysman Millstein Felder & Steiner LLP, 900 Third Avenue, New York, NY 10022 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WO 03/027891 A1

(54) Title: SYSTEM AND METHOD FOR ARCHIVING OBJECTS IN AN INFORMATION STORE

(57) Abstract: The invention relates generally to archiving data items in an information store. More particularly, the invention provides a computerized method for identifying, in a first information store, a first data item satisfying retention criteria; copying the first data item to a second information store; creating, in the first information store, a second data item containing a subset of the data of the first data item selected based on the data type of the first data item; and replacing the first data item, in the first information store, with the second data item.

SYSTEM AND METHOD FOR ARCHIVING OBJECTS IN AN INFORMATION STORE

PRIORITY CLAIM

This application claims priority from United States Provisional Patent Application No. 60/326,023, entitled "APPLICATION SPECIFIC OBJECT ARCHIVING AND RETRIEVAL SYSTEM", filed September 28, 2001. The entire contents of the Provisional Application 60/326,023 are hereby incorporated herein by reference in their entirety.

COPYRIGHT NOTICE

RELATED APPLICATIONS

This application is related to the following pending applications:

- Application Serial No. 09/610,738, titled MODULAR BACKUP AND RETRIEVAL SYSTEM USED IN CONJUNCTION WITH A STORAGE AREA NETWORK, filed July 6, 2000, attorney docket number 044463-002;
- Application Serial No. 09/609,977, titled MODULAR BACKUP AND RETRIEVAL SYSTEM WITH AN INTEGRATED STORAGE AREA FILING SYSTEM, filed August 5, 2000, attorney docket number 044463-0023;

- Application Serial No. 09/354,058, titled HIERARCHICAL BACKUP AND RETRIEVAL SYSTEM, filed July 15, 1999, attorney docket number 044463-0014;
- Application Serial No. 09/774,302, titled LOGICAL VIEW WITH GRANULAR ACCESS TO EXCHANGE DATA MANAGED BY A MODULAR DATA AND STORAGE MANAGEMENT SYSTEM, filed January 30, 2001, attorney docket number 044463-0040;
- Application Serial No.09/876,289, titled APPLICATION SPECIFIC ROLLBACK IN A COMPUTER SYSTEM, filed June 6, 2000, attorney docket number 044463-0029;
- Application Serial No. 09/774,272, titled EMAIL ATTACHMENT MANAGEMENT IN A COMPUTER SYSTEM, filed January 30, 2001, attorney docket number 4982/15;
- Application Serial No. 09/882,438, titled STORAGE OF APPLICATION SPECIFIC PROFILES CORRELATION TO DOCUMENT VERSIONS, filed June 14, 2001, attorney docket number 4982/16; and
- Application Serial No. _____, Titled COMBINED STREAM AUXILIARY COPY SYSTEM AND METHOD, filed September 16, 2002, attorney docket number 4982/26Prov;

each of which is hereby incorporated by reference in this application in its entirety.

BACKGROUND OF THE INVENTION

The invention disclosed herein relates generally to object archiving and retrieval in computer systems.

Electronic mail (e-mail) has increasingly become a common and accepted manner of exchanging messages for individuals both at home and in the workplace. Indeed, some e-mail users send and receive hundreds or even thousands of messages each day. Managing this large volume of message traffic, however, has become a problem for both individual users and network administrators.

When messages are sent and received by a mail application, they are stored for review in folders which are typically part of a file commonly referred to as an e-mail information store ("IS") that is designated to hold e-mail stored on the user's local computer or on a network storage device. Other types of applications such as directory services applications also have information stores which contain data specific to particular applications.

Over time, the IS typically grows in size as the user continues to receive and send more e-mail. This constantly increasing growth is problematic. Unless steps are periodically taken to reduce its size, the IS will eventually grow so large that it will use considerable amounts of disk space and also require excessive system resources to access its information. To keep the size of the IS under control and optimize system performance, administrators and users of e-mail systems have had to either delete or archive old or unwanted messages to release disk space. Both of these methods have serious drawbacks.

One problem associated with archiving old messages is that the archived messages are normally stored on the user's workstation in file formats such as .PST files which are difficult to manage. All references to individual messages

archived to .PST files no longer appear in the inbox and these individual messages are no longer readily accessible by browsing the e-mail client GUI. In order to review individual archived messages, users must know which archive contains their message and must open the individual archive containing the message before being able to access the message contents. This process is often time-consuming with users frequently resorting to trial-and-error methods of opening archives to locate desired messages.

Deleting old or unwanted messages is an even less desirable solution than archiving such messages. While archive files are difficult to manage and to retrieve messages from, deleting old or unwanted messages makes management and retrieval even more difficult and frequently impossible. If the user has performed a system backup prior to deleting such messages, retrieval is sometimes still possible, but the user must then restore the entire the entire system from the backup to retrieve the messages. In the worst case, the messages are simply lost forever when they are deleted.

Further, even in a networked environment with a central e-mail sever such as, for example, a Microsoft Exchange Server, which contains a central IS, the normal backup process will also not directly help cut down the size of the IS. Backing up the IS will still leave all of the messages in the IS unless the individual users delete or archive messages at their workstations.

There is thus a need for a system which permits users to easily manage archiving and retrieving e-mail messages.

In addition, similar problems relating to archiving of old or unwanted objects exist in other directory services applications such as Microsoft's Active Directory, the University of Michigan's LDAP Servers, Lotus Notes, Microsoft's

Sharepoint Portal, and other similar applications. In each of these applications, there exists a database similar to the Exchange IS which is constantly growing over time. System administrators must decide how much data in these databases is actually needed, how much should be archived, etc. One problem with archiving an entire directory services application database is that on restore, the entire database generally needs to be shut down even if only a small portion of the database needs to be restored. More single file restores are done than full system restores which results in inefficient use of system resources among other problems. There is thus also a need for a system which permits users to easily manage archiving and retrieving directory services and other similar application objects.

SUMMARY OF THE INVENTION

The present invention addresses the problems discussed above with the management of archiving and retrieving application specific archiving and retrieval.

In accordance with some aspects of the present invention, computerized methods are provided for archiving data, the methods comprising identifying, in a first information store, a first data item satisfying a retention criterion; copying the first data item from the first information store to a second information store; creating, in the first information store, a second data item containing a subset of the data of the first data item selected based on the data type of the first data item; and replacing the first data item, in the first information store, with the second data item. In some embodiments, the first data item may comprise an electronic mail message, an attachment to an electronic mail message, a directory services entry, or other data objects.

The retention criteria is a property or characteristic of the first data item used by the invention to select the first data item for archiving and other

purposes. In some embodiments, the retention criterion comprises a first creation date and identifying the first data item comprises comparing a first creation date of the first data item to the creation date specified as the retention criteria. In some embodiments, the retention criterion comprises a last accessed date and identifying the first data item comprises comparing a last accessed date of the first data item to the last accessed date specified as the retention criteria. In some embodiments, the retention criterion comprises an item size and identifying the first data item comprises comparing an item size of the first data item to the item size specified as the retention criteria.

In some embodiments, the first information store may comprise an electronic mail information store, a directory services information store, or other type of information store. In some embodiments, the second information store is a secondary storage device. In some embodiments, the second data item contains index information identifying a location of the first data item in the second information store.

In some embodiments, the second data item may comprise an electronic mail message, a directory services entry, or other type of data object. In some embodiments, the second data item contains header fields of the first data item which may include, for example, in the case of an electronic mail message, a sender field, a recipient field, a subject field, a date field, and a time field. In some embodiments, the second data item contains a subset of the message body of the first data item. In some embodiments, the second data item contains a message body specified by a user. In some embodiments, replacing the first data item comprises deleting the first data item from the first information store.

In one embodiment, the invention provides a system for archiving data, the system comprising a first information store containing one or more data items; a second information store; and a computer, connectable to the first information store and the second information store; wherein the computer is programmed to identify, in the first information store, a first data item satisfying a retention criteria; to copy the first data item to the second information store; to create, in the first information store, a second data item containing a subset of the data of the first data item selected based on the data type of the first data item; and to replace the first data item from the first information store. In some embodiments, the computer is programmed to identify an electronic mail message, an attachment to an electronic mail message, a directory services entry, and combinations thereof. In some embodiments, the computer is programmed to replace the first data item from the first information store by deleting the first data item.

In one embodiment, the invention provides a computer usable medium storing program code which, when executed on a computerized device, causes the computerized device to execute a computerized method for archiving data, the method comprising identifying, in a first information store, a first data item satisfying a retention criterion; copying the first data item from the first information store to a second information store; creating, in the first information store, a second data item containing a subset of the data of the first data item selected based on the data type of the first data item; and replacing the first data item, in the first information store, with the second data item.

In one embodiment, the invention provides a system for archiving data comprising a plurality of application-specific data agents each configured to coordinate archiving of first data items used in a particular software application; and a

plurality of application-specific stubbing modules each functionally integrated with a corresponding application-specific data agent, each stubbing module being configured to replace a first data item used in the corresponding software application with a second data item used in the corresponding software application and representing a subset of the first data item.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention is illustrated in the figures of the accompanying drawings which are meant to be exemplary and not limiting, in which like references are intended to refer to like or corresponding parts, and in which:

Fig. 1 is block diagram of a network architecture for a system to archive and retrieve application specific objects according to embodiments of the invention;

Fig. 2 is high-level flow chart of a method to archive application specific objects according to embodiments of the present invention;

Fig. 3 is a detailed flow chart of a method to archive application specific objects according to embodiments of the present invention;

Fig. 4 is a flow chart of a method to restore application specific objects according to embodiments of the present invention; and

Fig. 5 is an exemplary screen display of an e-mail message stub from a system to archive and retrieve application specific objects according to embodiments of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

Embodiments of methods and systems according to the present invention are described through references to Figs. 1 though 5. A network architecture for a system to archive and retrieve application specific objects in

accordance with embodiments of the present invention is depicted in Fig. 1. As shown, the system includes a storage manager 125 and one or more of the following: a data agent 105, a client computer 107, a first information store 108, a stubbing module 109, a media agent 110, a secondary storage library 115, and an index cache 120. The system and elements thereof are exemplary of a three-tier backup system such as the CommVault Galaxy backup system, available from CommVault Systems, Inc. of Oceanport, NJ, and further described in Application Number 09/610,738 which is incorporated herein by reference in its entirety.

A data agent 105 is a software module that is generally responsible for archiving, migrating, and recovering data of a client computer 107. Each client computer 107 has at least one data agent 105 and the system can support many client computers 107. The system provides a plurality of data agents 105 each of which is intended to backup, migrate, and recover data associated with a different application. For example, different individual data agents 105 may be designed to handle Microsoft Exchange data, Lotus Notes data, Microsoft Windows 2000 file system data, Microsoft Active Directory Objects data, and other types of data known in the art. If a client computer 107 has two or more types of data, one data agent 105 is required for each data type to archive, migrate, and restore the client computer 107 data. For example, to backup, migrate, and restore all of the data on a Microsoft Exchange 2000 server, the client computer 107 would use one Microsoft Exchange 2000 Mailbox data agent 105 to backup the Exchange 2000 mailboxes, one Microsoft Exchange 2000 Database data agent 105 to backup the Exchange 2000 databases, one Microsoft Exchange 2000 Public Folder data agent 105 to backup the Exchange 2000 Public Folders, and one Microsoft Windows 2000 File System data agent 105 to backup the client computer's 107 file system. These data agents 105 are addressed as

four separate data agents 105 by the system even though they reside on the same client computer 107.

The data stubbing module 109 is a component of the media agent that performs stubbing operations on data items as further described herein.

A media agent 110 conducts data between the client computer 107 and one or more storage libraries 115 such as a tape library or other storage device. The media agent 110 is communicatively coupled with and controls the storage library 115. For example, the media agent 110 might instruct the storage library 115 to use a robotic arm or other means to load or eject a media cartridge, and to archive, migrate, or restore application specific data. The media agent 110 generally communicates with the storage library 115 via a local bus such as a SCSI adaptor. In some embodiments, the storage library 115 is communicatively coupled to the data agent 110 via a Storage Area Network ("SAN").

Each media agent 110 maintains an index cache 120 which stores index data the system generates during backup, migration, and restore storage operations as further described herein. For example, storage operations for Microsoft Exchange data generate index data. Index data is useful because it provides the system with an efficient mechanism for locating user files for recovery operations. Although this index data is generally stored with the data backed up to the storage library 115, the media agent 110 that controls the storage operation also writes an additional copy of the index data to its index cache 120. The data in the index cache 120 is thus readily available to the system for use in storage operations and other activities without having to be first retrieved from the storage library 115.

Each index cache 120 typically resides on the corresponding media agent's 110 hard disk or other fixed storage device. Like any cache, the index cache

120 has finite capacity and the amount of index data that can be maintained directly corresponds to the size of that portion of the disk that is allocated to the index cache 120. In one embodiment, the system manages the index cache 120 on a least recently used ("LRU") basis as known in the art. When the capacity of the index cache 120 is reached, the system overwrites those files in the index cache 120 that have been least recently accessed with the new index data. In some embodiments, before data in the index cache 120 is overwritten, the data is copied to the index cache 120 copy in the storage library 115. If a recovery operation requires data that is no longer stored in the index cache 120 such as in the case of a cache miss, the system recovers the index data from the index cache 120 copy stored in the storage library 120.

The storage manager 125 is a software module or application that coordinates and controls the system. The storage manager 125 communicates with all elements of the system including media agents 110, client computers 107, and data agents 105 to initiate and manage system backups, migrations, and recoveries.

In some embodiments, components of the system may reside and execute on the same computer. In some embodiments, a client computer 107 component such as a data agent 105, a media agent 110, or a storage manager 125 coordinates and directs local archiving, migration, and retrieval application functions as further described in Application Number 09/610,738. This client computer 107 component can function independently or together with other similar client computer 107 components.

Fig. 2 presents high-level flow chart of a method to archive application specific objects in accordance with embodiments of the present invention. An archive job automatically starts, step 130, according to a pre-defined schedule or as manually directed by a user. In some embodiments, the storage manager 125 instructs the

media agent 110 to begin an archive job and the media agent 110 then instructs the data agent 105 to commence an archive process. In other embodiments, the media agent 110 directly instructs the data agent 105 to commence an archive process without instructions from the storage manager 125.

The archive process filters and archives those messages, objects, or other data items in a first information store 108 according to specified retention criteria, step 135. In some embodiments, the retention criteria may be input directly by a user when the archive job is started. In other embodiments, the retention criteria may be pre-defined or calculated automatically according to user preferences or other information and the archive job proceeds autonomously. Those messages, objects, or other data items that fulfill the specified retention criteria are copied from the first information store 108 to a second information store. In some embodiments, the second information store is located on secondary storage media such as a storage library 115.

A stubbing process creates a stub for and deletes each message in the first information store 108 that was copied to the second information store, step 140. As used herein, stubs are data objects that replace messages, objects, and other data items in the first information store 108 that were copied to the second information store. Copying the messages to the second information store frees storage space in the first information store 108. Stubs that replace the copied messages generally require less storage space and indicate which items were copied and deleted from the first information store 108. Stubs also facilitate retrieval of messages that were copied to the second information store.

When all items in the first information store 108 have been archived and stubbed or when directed by a user, the job ends, step 145.

Fig. 3 presents detailed flow chart of a method to archive application specific objects in accordance with embodiments of the present invention. A job manager starts an archiving job beginning with an archiving phase and notifies an archive management daemon on the client computer 107, step 150. The job manager is a software process that is generally a part of the storage manager 125, but in some embodiments the job manager may also be part of the media agent 110, the data agent 105, or any combination thereof. In some embodiments, the system starts with the archiving phase to ensure that messages, objects, or other data items are only stubbed after they are backed-up to secondary storage. Those skilled in the art will recognize that multiple archiving jobs could be run at one time.

As previously described herein, archive jobs can either be started manually as directed by a user or automatically as a scheduled system management task. In some embodiments, archive jobs may take place according to schedule policies as further described in Application No. 09/882,438 which is incorporated herein by reference in its entirety. For example, a schedule policy may indicate that archive jobs should take place on a specified information store once per day at a particular hour or at other designated times. In some embodiments, the archiving process and stubbing process can also be scheduled at off-peak times to reduce the load to system resources.

The archive management daemon initiates an archiving process of the data agent 105, step 155, which archives messages, objects, or other data items in a first information store 108 according to specified retention criteria.

The archiving process scans an item in the first information store 108 to determine whether the item fulfills the retention criteria, step 160. For example, in an e-mail information store, the archiving process scans the mailboxes in the IS 108 to

find candidate messages or objects meeting either the default retention criteria, such as the retention criteria specified in a storage policy, or the retention criteria customized by the user.

As previously described, retention criteria define archiving rules for the archiving process to control the content of stubs, which messages, objects, or other data items get archived, the retention time for stubs and archived messages, objects, or other data items, and other similar filtration criteria. In one embodiment, messages, objects, and other data items are copied to secondary storage according to parameters such as job ID, times, etc. In other embodiments, retention criteria specify additional options to indicate whether a stub should be left behind or not, whether the entire message or object or just the attachment should be archived, and other similar choices. In some embodiments, retention criteria specify a mailbox size threshold and exclusion filter for mailbox(es) or folder(s), so that only mailboxes whose size exceed the threshold and are not in the filter list will be scanned. In some embodiments, retention criteria also specify rules based on message creation time, modification time, size, or attachment(s) to further control the message selection criteria. For example, messages in the IS 108 that satisfy certain retention criteria such as age, size, size of attachments, amount of disk space left, size of mailbox, etc. are archived to secondary storage 115.

Since stubs are usually new small messages or objects with no attachments, they can be difficult to remove from a users mailbox. To facilitate stub management among other things, retention criteria also define the lifetime of a stub in some embodiments. After a stub expires past its lifetime, the next archiving job will either delete the stub from the first information store 108 or archive the stub to secondary storage such as a storage library 115.

The size of index cache 120 may grow over time and in some embodiments, archive pruning-related retention criteria specifies that data should be pruned or deleted in the first information store 108 and also in the index cache 120. In some embodiments, retention criteria may also specify whether archived messages, objects, and other data items in secondary storage 115 should be pruned after additional time has passed or at any desired point.

In some embodiments, retention criteria specifies that there should be no orphan stubs left in the IS 108. In one embodiment, this goal among others is achieved by using retention times, such that stubs always expire before their related messages, objects, or other data items archived in secondary storage 115. In other embodiments, retention criteria specifies that items archived in secondary storage 115 are not pruned if a stub still exists in the first information store 108. In an alternate embodiment, archive pruning of secondary storage 115 items produces a pruned list stored in the index cache 115 or other information store and the system uses this list to then remove the related stubs remaining in the first information store 108. In some embodiments, however, retention criteria may permit messages archived in secondary storage to be pruned even if a related stub still exists in the first information store 108.

If the item scanned fulfills the retention criteria, step 165, the archiving process writes a copy of the message, object, or other data item to secondary storage 115, step 170, as further described in Application No. 09/610,738 and Application No. 09/609,977 which are incorporated herein by reference in their entirety. To make the restore process faster and to achieve other desirable goals, messages, objects, and other data items can first be archived on a magnetic library. Later these items can be moved to tape or some other storage media for long-term storage. In one embodiment, data may be moved from a magnetic library to tape or some other

secondary storage media using auxiliary copy to further conserve system resources, and as described in Application No. _____, COMBINED STREAM AUXILIARY COPY SYSTEM AND METHOD, filed September 16, 2002, attorney docket number 4982/26Prov, which is incorporated herein by reference in its entirety.

Identifying characteristics and other information about the item copied to secondary storage 115 are recorded to an archiving list stored in a local information store of the data agent 105, step 175. During the archiving phase, a record detailing information about every item successfully copied from the first information store 108 to secondary storage 115 will be stored in the archiving list which serves as input and is used during the stubbing phase as further described herein. The content of items in the archiving list include information for the later stubbing phase and restore process to locate the original archived messages, objects, or other data items. Examples of such information include mailbox name, folder ID, message ID, and link information to the item's secondary storage location. An example of such link information is a Universal Naming Convention ("UNC") path to the item's index entry in a Galaxy file system.

The system determines whether additional items remain in the first information store 108 to be scanned against the retention criteria, step 180. If additional items remain to be scanned, then control returns to step 160, and the process repeats. Otherwise, the archiving process terminates and the job manager then starts the archiving index phase writing the archive information to the index cache 115 on a media agent 110 or other component, step 185, as further described in Application Number 09/610,738 which is hereby incorporated by reference in its entirety.

When the archiving index phase is complete, the media agent notifies the job manager which then initiates a stubbing process, step 190. The stubbing process retrieves the archiving list of messages, objects, or other data items created during the archiving process to use as input during the stubbing phase and sequentially processes each item on the archiving list, step 195. While the stubbing process could query the media agent 110 containing the index cache 115 created during the archiving index phase, this option is less desirable due to the network and processor resources required. A more desirable method, as described herein, combines archiving and stubbing into a single job in which the stubbing phase only starts after the archiving phase is successfully completed. In one embodiment, the stubbing phase processes items on the archiving list based on application ID, job status, and other criteria.

In some embodiments, before a stub is created, the system prompts for, step 200, and accepts, step 205, user input of text and other information to display or otherwise associate with a stub.

For each item on the archiving list, the stubbing process then creates a new "stub" message, object, or other data item in the first information store 108, step 210. Each new message has the same data structure as the original message, except the body field of the message is replaced with explanation text or other information indicating that the message is a stub, and a path linking to the archived message.

New stubs are generally created according to stub configuration options specified in the storage policy associated with the first information store 108. Stub configuration options include stub with no body or stub with full body, but no attachment, etc. The subject of the stub can be changed to incorporate a tag or other parsable identifier such as "<archived> original subject" so that subsequent archive

operations can identify the stub. In some embodiments, there are also named properties in stubs. In some embodiments, stubs contain an ID and an archive time to assist backup systems such as Galaxy with stub management.

After each stub is successfully created, the stubbing process deletes the original message, step 215, and determines whether there are additional items to process on the archiving list, step 220. If additional items remain to be processed, control returns to step 195. Otherwise, once all messages, objects, and other data items on the archiving list have been processed, the stubbing process returns control to the job manager and the archiving job terminates, step 225.

Fig. 4 presents a flow chart of a method to restore application specific objects according to embodiments of the present invention. The message, object, or other data item to restore is selected, step 230. The item may be selected automatically by the system according to predefined restore criteria stored in the index cache 115, in a storage policy, or in another information store. The system also accepts user input to manually select the item to restore.

Turning to Fig. 5, for example, a sample screen display shows an e-mail message stub from a system to archive and retrieve application specific objects according to embodiments of the present invention. As shown, the stub includes many of the header fields of the original archived message including the sender 260, the recipient 265, the subject 270, and the time/date 275. The stub also includes a message 280 indicating that the body of the original message has been archived, a link 285 to the archived message body, a message 290 indicating that an attachment to the original message has been archived, and a link 295 to the archived attachment.

In one embodiment, items archived in secondary storage, such as the message body, can be restored by manually clicking on the UNC link 285 or other

identifying path in the item's related stub. The stub message ID is encoded within the link 285. The media agent 210 or the data agent 205 detects the click, parses the message ID, and passes the ID as a parameter to the restore process, as further described herein.

If an archived e-mail message's corresponding stub is still in a mailbox or other browsable first information store 108, a user can search on the stubs' fields copied from the archived e-mail message including the sender 260, recipient 265, subject 270, time/date 275, and other identifying criteria to find the corresponding stub. In some embodiments, stubs with full bodies can also be located via full-text (index) searching functions of a mail server or other file system.

Sometimes, stubs will no longer be accessible via the first information store 108. For example, stubs may have been pruned or otherwise deleted. In another embodiment, the archived message can be selected via the archived message's index entry in a secondary storage file system such as in the Galaxy system. For example, if the user wants to restore an archived e-mail message whose corresponding stub has been pruned from the first information store 108 mailbox, the user can still restore the archived e-mail via a backup system console browser such as the Galaxy console browser.

As further described in Application No. 09/610,738 and Application No. 09/609,977 which are incorporated herein by reference in their entirety, selecting an item to restore triggers the media agent 210 mapped storage installable file system, and the media agent 210 sends a restore request to a job manager process on the media agent, step 235. The job manager on the media agent 210 starts a restore job using the information contained in the request and notifies a job manager process of the data agent 205, step 240. The job manager process of the data agent 205 creates a

restore process which retrieves the archived message from the secondary storage library 115 returning it to the first information store 108, step 245. After the archived item is restored from secondary storage 115 and copied to the first information store 108, the item's corresponding stub is deleted from the first information store 108, step 250, and the job ends, step 255.

In some embodiments, users are prevented from modifying stubs since the restore process depends on special information contained in each stub to identify it as a stub and to restore the original message.

In one embodiment, if users are moved to another mail server having a different information store than the first information store 108, the system first restores all the stubbed messages, objects, and other data items back to the user's mailbox in the first information store 108, and then deletes all the stubs before the administrator is permitted to move the user. In some embodiments, this is accomplished as an integrated part of the system or as a separate process to scan the mailbox in the first information store 108, find all the stubs, pass the links to the media agent 119 to start a restore job, and then delete the stubs. In some embodiments, stubs contain application type, backup management system console information such as CommVault CommServer information, and other information which ensures compatibility and continued functionality of the invention if a user updates their mail server.

While the system has frequently been described above in the context of electronic mail object archiving, the system also archives and restores general directory services client objects for each entry that exists in a service such as Microsoft's Active Directory, the University of Michigan's LDAP Servers, Lotus Notes, Microsoft's Sharepoint Portal, and other similar applications. Attributes and

properties of each archived service entry are retained in the corresponding stub. Client operations are performed using an interface such as the Windows LDAP API, which can interface with Active Directory, as well as any other LDAP compliant directory service. Similarly, the directory services client looks and behaves very much like an e-mail file system agent from a GUI standpoint with backup sets and sub-clients where default sub-clients backup the entire directory service and new sub-clients are defined to limit the scope of the backup. Filters specify retention criteria to archive certain branches of the directory tree, certain entries, and certain attributes. Each entry is stored in a separate file that adheres to the interface format, such as the LDIF (LDAP Data Interchange Format) format, which is an RFC standard format for listing the attributes of an entry.

Systems and modules described herein may comprise software, firmware, hardware, or any combination(s) of software, firmware, or hardware suitable for the purposes described herein. Software and other modules may reside on servers, workstations, personal computers, computerized tablets, PDAs, and other devices suitable for the purposes described herein. Software and other modules may be accessible via local memory, via a network, via a browser or other application in an ASP context, or via other means suitable for the purposes described herein. Data structures described herein may comprise computer files, variables, programming arrays, programming structures, or any electronic information storage schemes or methods, or any combinations thereof, suitable for the purposes described herein. User interface elements described herein may comprise elements from graphical user interfaces, command line interfaces, and other interfaces suitable for the purposes described herein. Screenshots presented and described herein can be displayed

differently as known in the art to input, access, change, manipulate, modify, alter, and work with information.

While the invention has been described and illustrated in connection with preferred embodiments, many variations and modifications as will be evident to those skilled in this art may be made without departing from the spirit and scope of the invention, and the invention is thus not to be limited to the precise details of methodology or construction set forth above as such variations and modification are intended to be included within the scope of the invention.

WHAT IS CLAIMED IS:

1. A computerized method for archiving data, the method comprising:

identifying, in a first information store, a first data item satisfying a retention criterion;

copying the first data item from the first information store to a second information store;

creating, in the first information store, a second data item containing a subset of the data of the first data item selected based on the data type of the first data item; and

replacing the first data item, in the first information store, with the second data item.

2. The method of claim 1, wherein identifying the first data item comprises identifying an electronic mail message.

3. The method of claim 1, wherein identifying the first data item comprises identifying an attachment to an electronic mail message.

4. The method of claim 1, wherein identifying the first data item comprises identifying a directory services entry.

5. The method of claim 1, wherein the retention criterion comprises a first creation date and wherein identifying the first data item satisfying the retention criterion comprises comparing a first creation date of the first data item to the creation date.

6. The method of claim 1, wherein the retention criterion comprises a last accessed date and wherein identifying the first data item satisfying the retention criterion comprises comparing a last accessed date of the first data item to the last accessed date.

7. The method of claim 1, wherein the retention criterion comprises an item size and wherein identifying the first data item satisfying a retention criteria comprises comparing an item size of the first data item to the item size.

8. The method of claim 1, wherein the first information store comprises an electronic mail information store and wherein identifying the first data item comprises identifying an electronic mail message or an attachment to an electronic mail message.

9. The method of claim 1, wherein the first information store comprises a directory services information store and wherein identifying the first data item comprises identifying a directory services entry.

10. The method of claim 1, wherein copying the first data item from the first information store to a second information store comprises copying the first data item from the first information store to a secondary storage device.

11. The method of claim 1, wherein creating the second data item comprises creating, in the first information store, the second data item containing index information identifying a location of the first data item in the second information store.

12. The method of claim 1, wherein creating the second data item comprises creating an electronic mail message.

13. The method of claim 12, wherein creating the electronic mail message comprises creating, in the first information store, an electronic mail message containing header fields of the first data item.

14. The method of claim 13, wherein the header fields include one or more from the group consisting of: a sender field, a recipient field, a subject field, a date field, and a time field.

15. The method of claim 12, wherein creating the electronic mail message comprises creating, in the first information store, an electronic mail message containing a subset of the message body of the first data item.

16. The method of claim 12, wherein creating the electronic mail message comprises creating, in the first information store, an electronic mail message containing a new message body specified by a user.

17. The method of claim 1, wherein creating the second data item comprises creating, in the first information store, a directory services entry.

18. The method of claim 1, wherein replacing the first data item comprises deleting the first data item from the first information store.

19. A system for archiving data, the system comprising:

a first information store containing one or more data items;

a second information store; and

a computer, connectable to the first information store and the second information store;

wherein the computer is programmed to identify, in the first information store, a first data item satisfying a retention criteria; to copy the first data item to the second information store; to create, in the first information store, a second data item containing a subset of the data of the first data item selected based on the data type of the first data item; and to replace the first data item from the first information store.

20. The system of claim 19, wherein the computer is programmed to identify an electronic mail message.

21. The system of claim 19, wherein the computer is programmed to identify an attachment to an electronic mail message.

22. The system of claim 19, wherein the computer is programmed to identify a directory services entry.

23. The system of claim 19, wherein the computer is programmed to replace the first data item from the first information store by deleting the first data item.

24. A computer usable medium storing program code which, when executed on a computerized device, causes the computerized device to execute a computerized method for archiving data, the method comprising:

identifying, in a first information store, a first data item satisfying a retention criterion;

copying the first data item from the first information store to a second information store;

creating, in the first information store, a second data item containing a subset of the data of the first data item selected based on the data type of the first data item; and

replacing the first data item, in the first information store, with the second data item.

25. A system for archiving data comprising:

a plurality of application-specific data agents each configured to coordinate archiving of first data items used in a particular software application; and

a plurality of application-specific stubbing modules each functionally integrated with a corresponding application-specific data agent, each stubbing module being configured to replace a first data item used in the corresponding software application with a second data item used in the corresponding software application and representing a subset of the first data item.

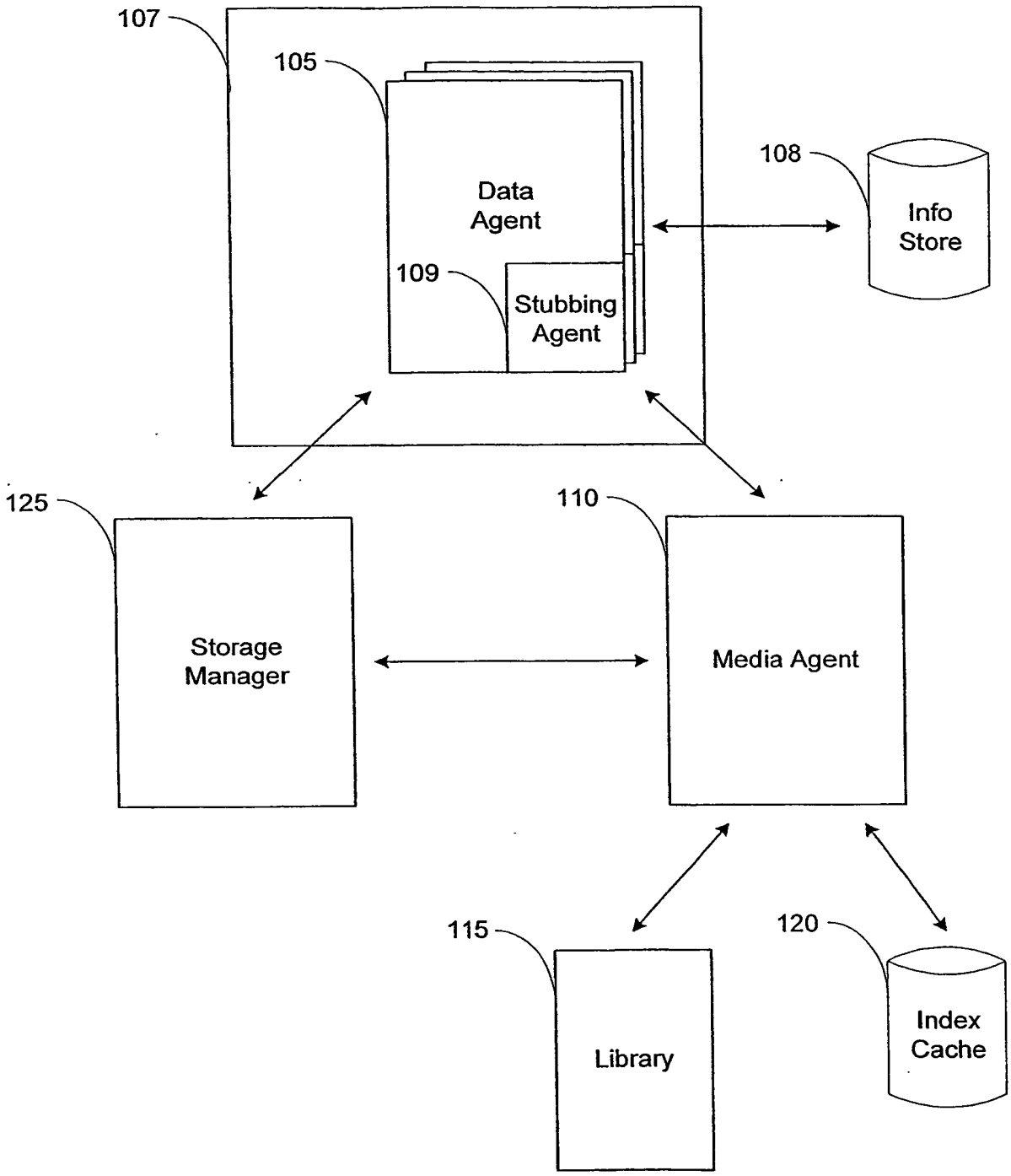
107
105
Data Agent
109
Stubbing Agent
108
Info Store
125
Storage Manager
110
Media Agent
115
Library
120
Index Cache

Fig. 1

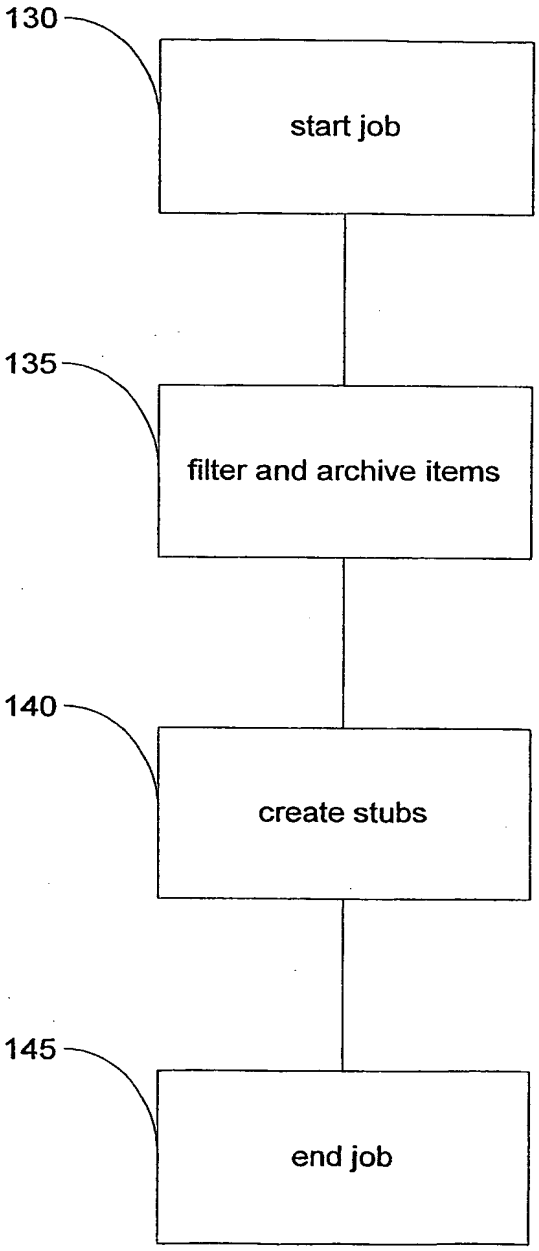
130
start job
135
filter and archive items
140
create stubs
145
end job

Fig. 2

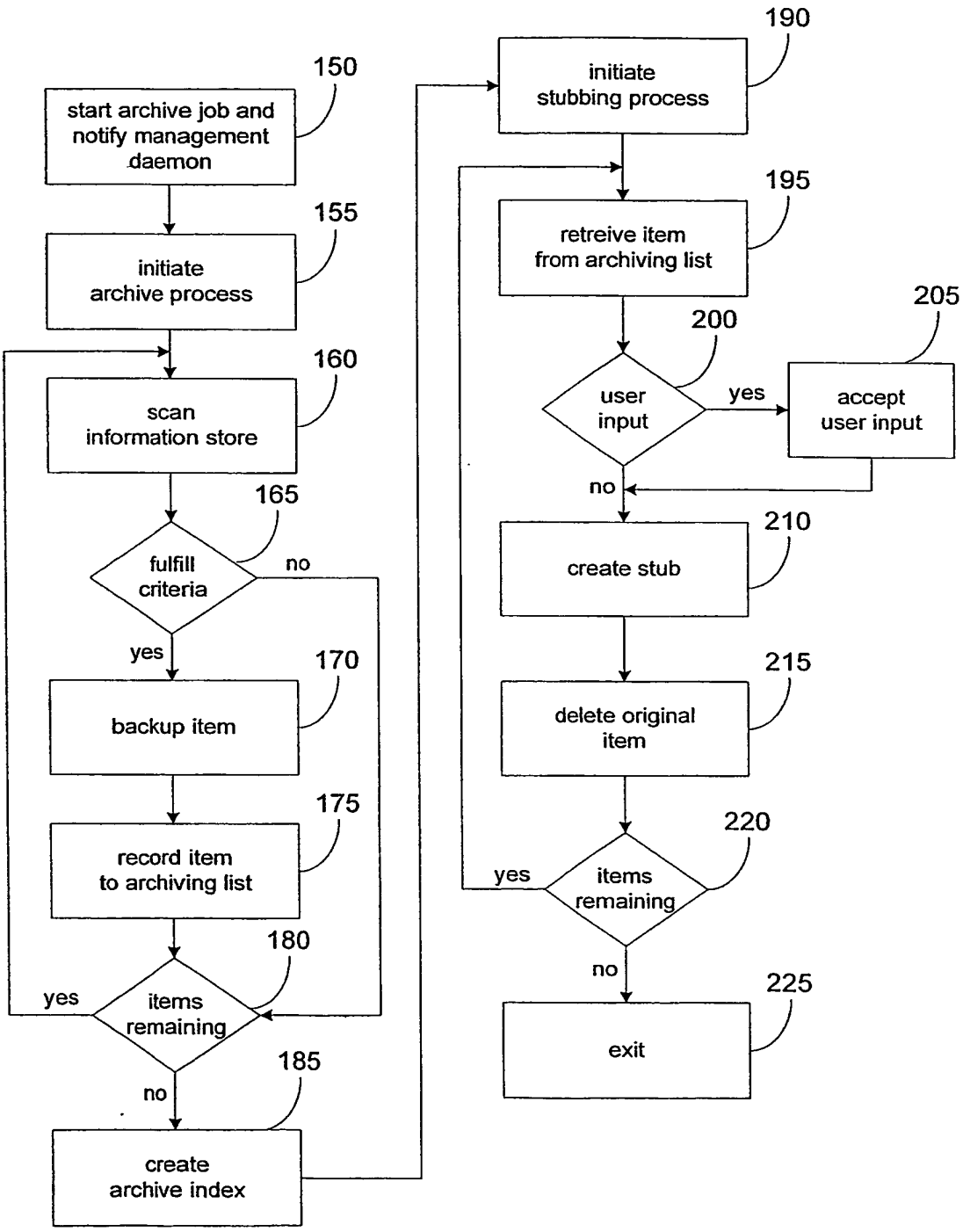
150
start archive job and notify management daemon
155
initiate archive process
160
scan information store
165
fulfill criteria
no
yes
170
backup item
175
record item to archiving list
180
items remaining
yes
no
185
create archive index
190
initiate stubbing process
195
retreive item from archiving list
200
user input
yes
205
accept user input
no
210
create stub
215
delete original item
220
items remaining
yes
no
225
exit

Fig. 3

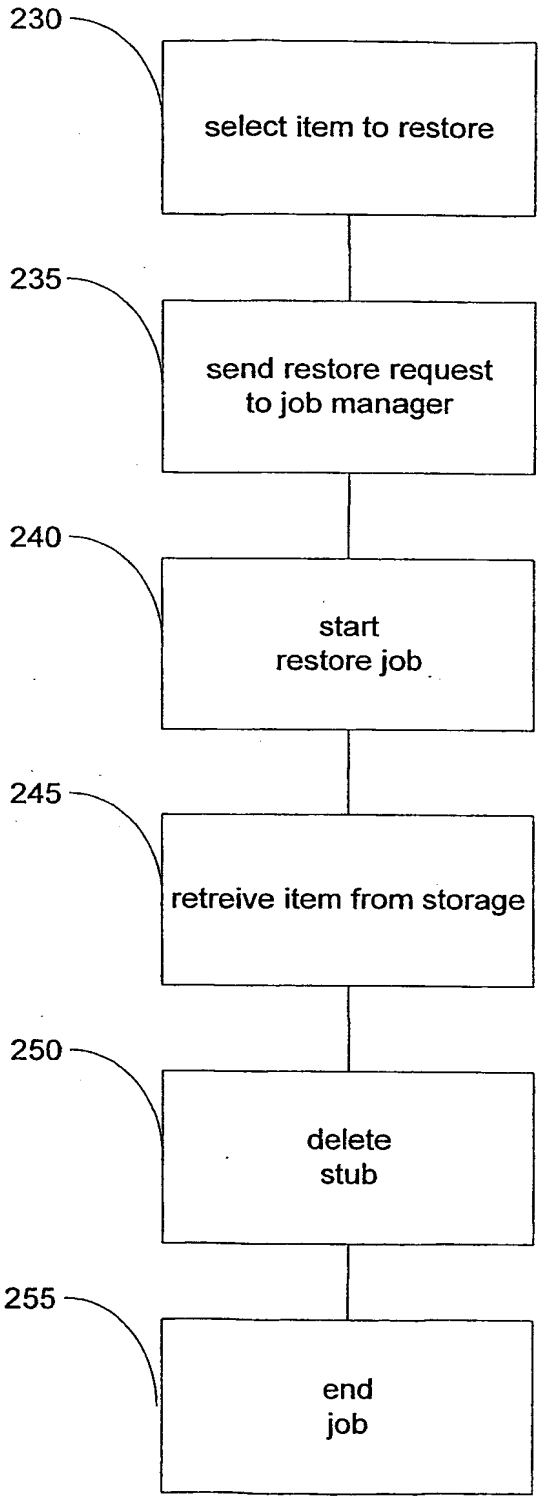
230
select item to restore
235
send restore request to job manager
240
start restore job
245
retreive item from storage
250
delete stub
255
end job

Fig. 4

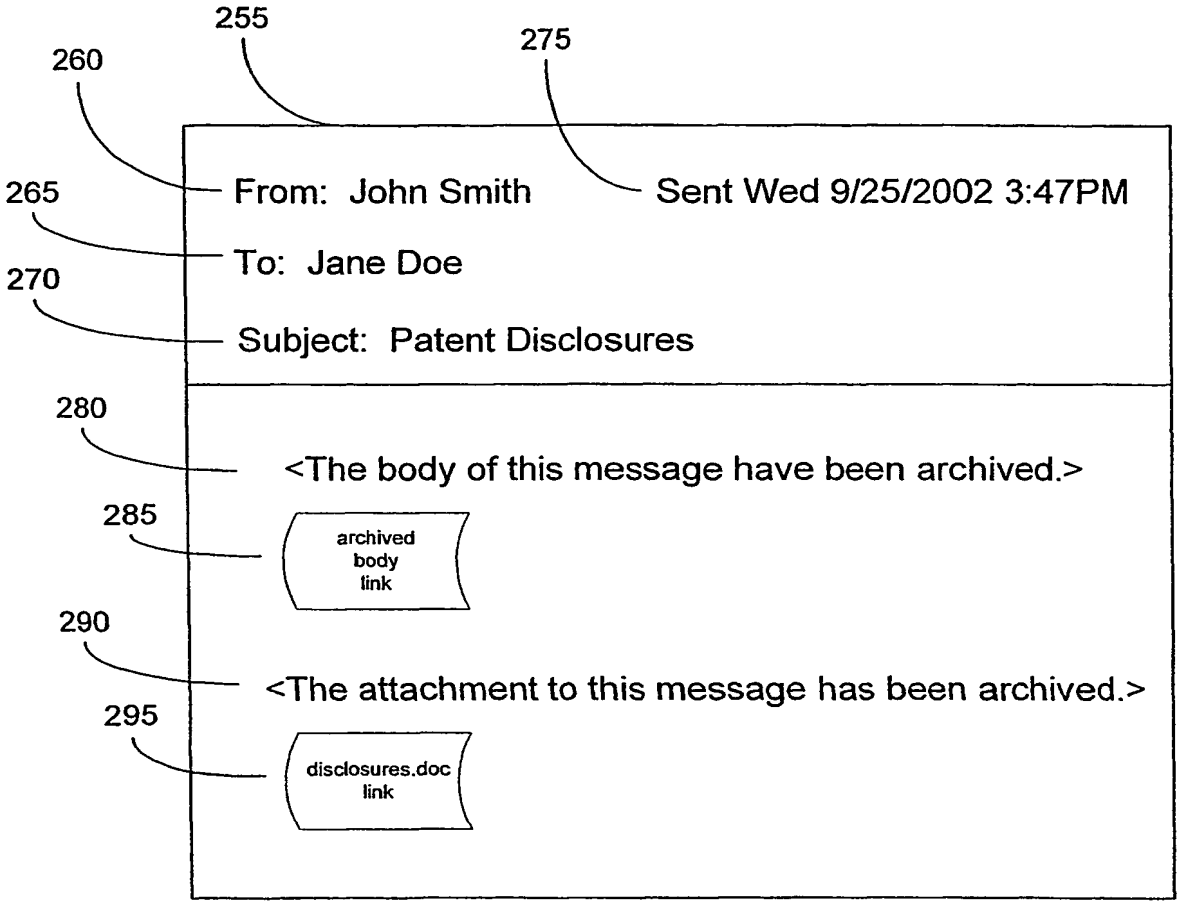
255
260
275
From: John Smith
Sent Wed 9/25/2002 3:47PM
265
To: Jane Doe
270
Subject: Patent Disclosures
280
<The body of this message have been archived.>
285
archived body link
290
<The attachment to this message has been archived.>
295
disclosures.doc link

Fig. 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US02/31205

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) : G06F 15/173
US CL : 707/204, 709/206

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
U.S. : Please See Continuation Sheet

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X --- Y	US 6,026,414 A (ANGLIN) 15 February 2000 (15.02.2000), Column 1, lines 11-26; Column 2, lines 30-56; col. 4, lines 43-54; Figures, 1-3.	1, 4, 19, 22, 24-25 --- 2-3, 5-18, 20-21, 23
Y,P	US 6,327,612 B1 (WATANABE) 04 December 2001 (04.12.2001), Abstract; Column 1, line 61 - Column 2, line 51; Column 3, lines 15-37; Column 4, lines 15-42, lines 50-52; Column 4, line 64 - Column 5, line 11; lines 32-45; Figures 1-3, 5-8.	2-3, 5-18, 20-21, 23
A,P	US 6,345,288 B1 (REED et al.)05 February 2002 (05.02.2002), All	1-25

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* *Special categories of cited documents:*

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
04 December 2002 (04.12.2002)	24 DEC 2002
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703)305-3230	Authorized officer for Te Y Chen Telephone No. (703)308-6296

Form PCT/ISA/210 (second sheet) (July 1998)

INTERNATIONAL SEARCH REPORT

PCT/US02/31205

Continuation of B. FIELDS SEARCHED Item 1:
707/9-10, 200, 204, 709/205-206, 232-238

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.